

Programas usando el lenguaje ensamblador

Ejemplo 1. Realizar un programa que genere un contador a partir del numero 7 y se incremente de uno en uno. El resultado del contador debe colocarse en la dirección de memoria 5. Observemos el diagrama de flujo de la figura 1.

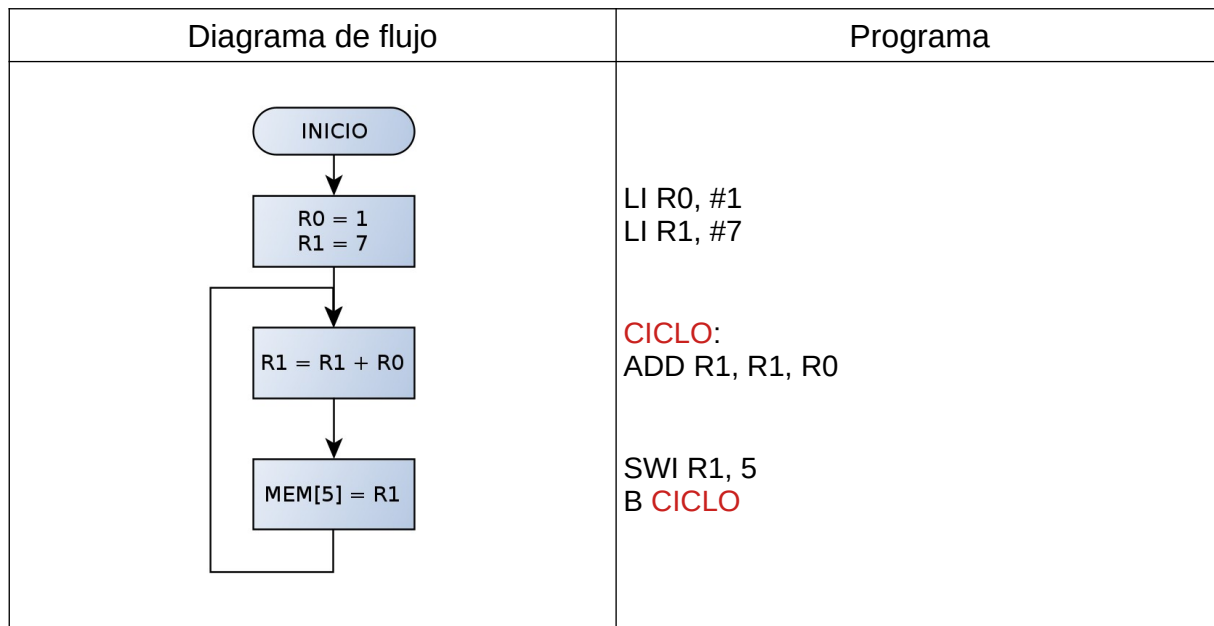


Figura 1: Ejemplo1 - contador

El programa que implementa el diagrama de flujo de la figura 1 se muestra en la tabla 1.

Instrucciones	Significado	Dir	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	T
LI R0, #1	R0 = 1	0							00									I
LI R1, #7	R1 = 7	1							01									I
CICLO: ADD R1, R1, R0	R1 = R1 + R0	2							01	01	00							R
SWI R1, 5	Mem[5] = R1	3							01									I
B CICLO	PC = 2	4							xx									J

Tabla 1: Formatos de instrucción del programa del ejemplo 1

Ejemplo 2. Realizar un programa que obtenga los primeros **12 términos** de la serie de Fibonacci. Los valores iniciales de la serie son 0 y 1 y se colocan en R0=0 y R1=1. Cada término de la serie se obtiene sumando los 2 números anteriores. Además, el término de la serie calculado debe colocarse en la dirección de memoria 72. Observemos el diagrama de flujo de la figura 2.

1er término	2o término	3er término	4o término	5o término	6o término	7o término	8o término ...
0	1	1	2	3	5	8	13 ...
R0	R1	R0	R1	R0	R1	R0	R1 ...

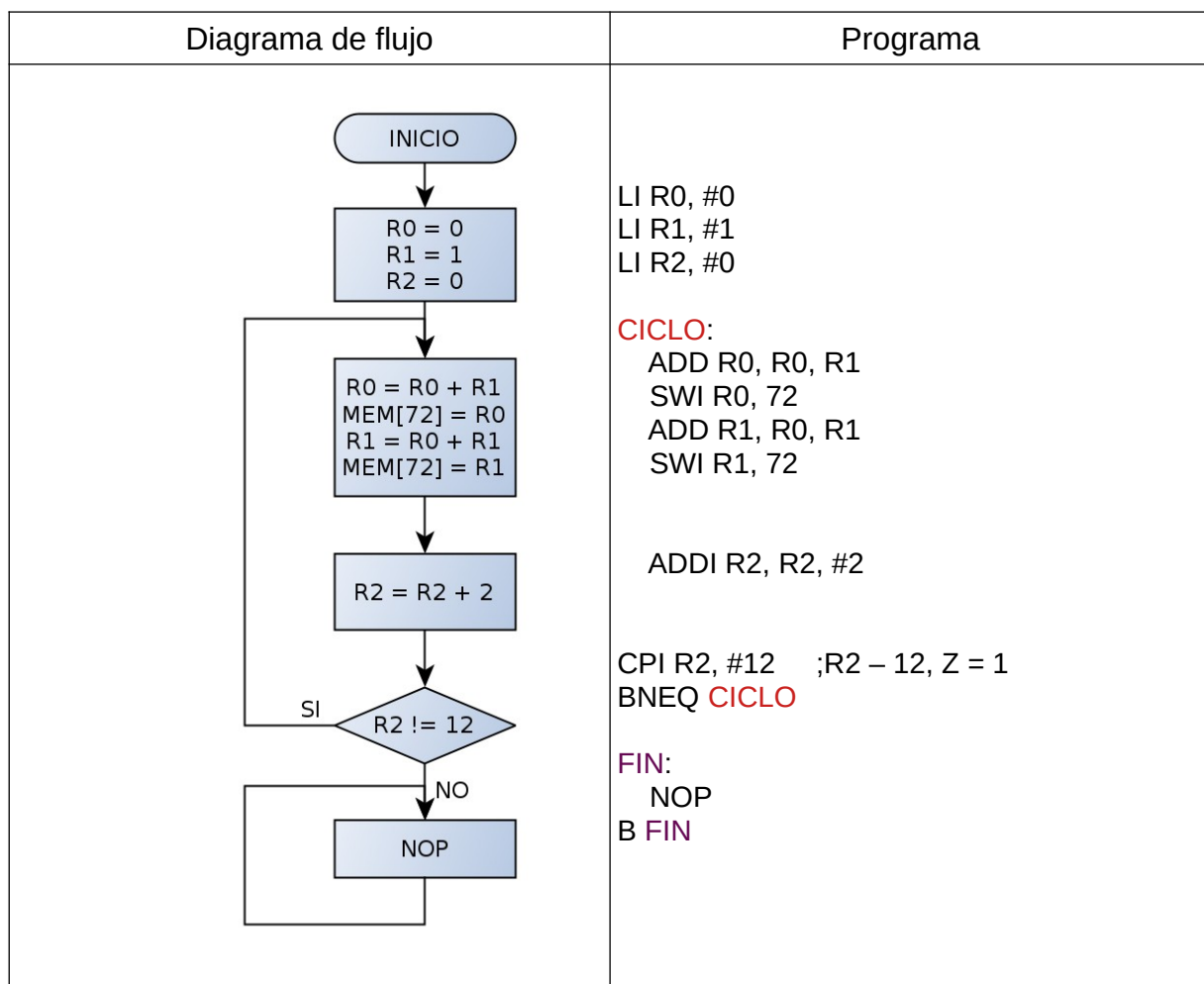


Figura 2: Ejemplo 2- Serie de Fibonacci

El programa que implementa el diagrama de flujo de la figura 2 se muestra en la tabla 2.



Instrucciones	Significado	Dir	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	T
LI R0, #0	R0 = 0	0		00001				00				00000000						I
LI R1, #1	R1 = 1	1		00001				01				00000001						I
LI R2, #0	R2 = 0	2		00001				10				00000000						I
CICLO: ADD R0, R0, R1	R0 = R0 + R1	3		00000				00		00		01		0000				R
SWI R0, 72	Mem[72] = R0	4		00011				00				01001000						I
ADD R1, R0, R1	R1 = R0 + R1	5		00000				01		00		01		0000				R
SWI R1, 72	Mem[72] = R0	6		00011				01				01001000						I
ADDI R2, R2, #2	R2 = R2 + 2	7		00100				10		10		000010						I
CPI R2, #12	R2 - 12	8		00111				10				00001100						I
BNEQ CICLO	PC=(Z==0)?3:10	9		01001				XX				00000011						J
FIN: NOP		10		01110				XX		XX		XX		XXXX				
B FIN	PC = 10	11		00110				XX				00001010						J

Tabla 2: Formatos de instrucción del programa del ejemplo 2

Ejemplo 3. Realizar un programa que genere un contador a partir del número **cero** y se incremente de uno en uno. Determinar si el resultado del contador es un número par o impar. Si el número es par, colocar el código F0H en la dirección de memoria AH, si es impar colocar el código 0FH en la misma dirección. Observemos el diagrama de flujo de la figura 3.

Número par = 10

	7	6	5	4	3	2	1	0
Número = 10	0	0	0	0	1	0	1	0
Máscara = 1	0	0	0	0	0	0	0	1
Resultado AND	0	0	0	0	0	0	0	0

Número impar = 13

	7	6	5	4	3	2	1	0
Número = 13	0	0	0	0	1	1	0	1
Máscara = 1	0	0	0	0	0	0	0	1
Resultado AND	0	0	0	0	0	0	0	1

Para identificar si un número es par o impar se realiza una operación lógica AND entre el número a verificar y la máscara 0x01, si el resultado de la operación lógica es CERO, se trata de un número PAR, por el contrario, si resultado de la operación lógica es UNO, se trata de un número IMPAR.

AUTOR: VICTOR HUGO GARCÍA ORTEGA

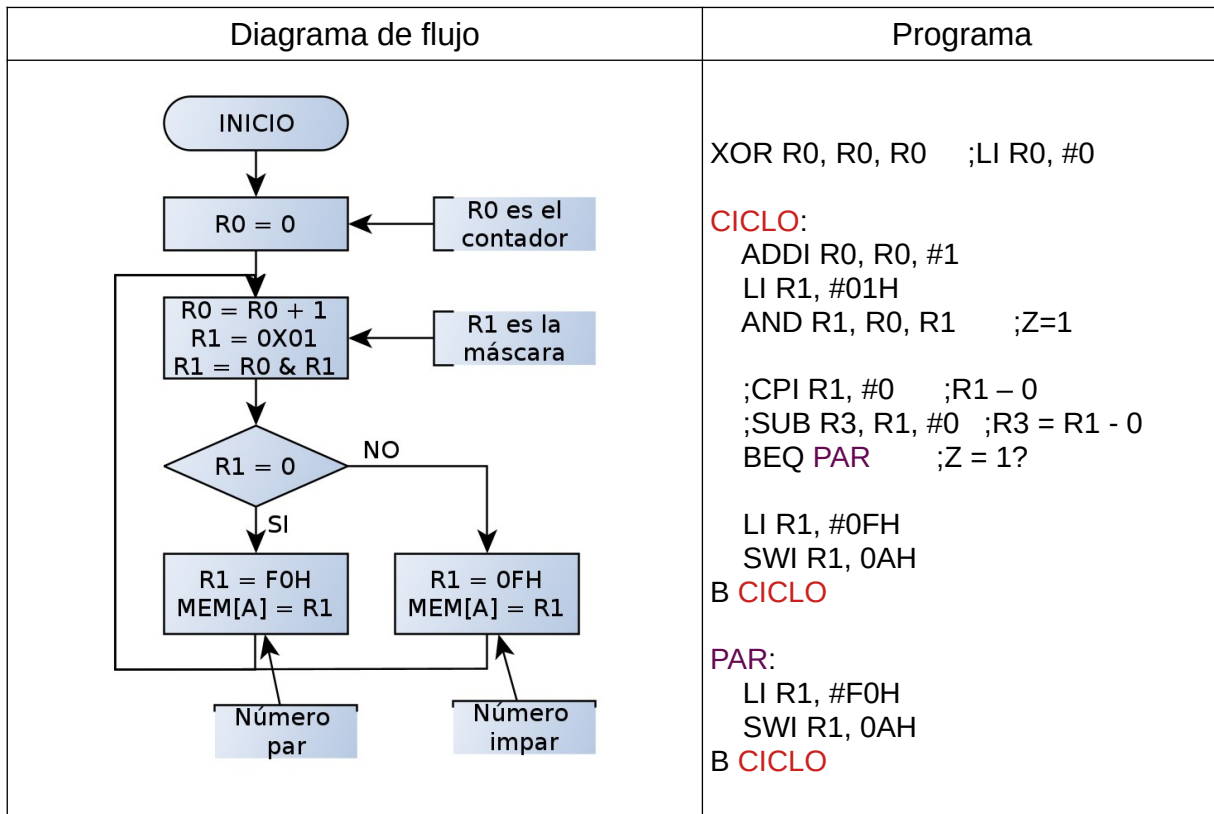


Figura 3: Ejemplo3 - Números pares e impares

El programa que implementa el diagrama de flujo de la figura 3 se muestra en la tabla 3.

Instrucciones	Significado	Dir	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	T
XOR R0, R0, R0		0																
CICLO: ADDI R0, R0, #1		1																
LI R1, #01H		2																
AND R1, R0, R1		3																
BEQ PAR		4																
LI R1, #0FH		5																
SWI R1, 0AH		6																
B CICLO		7																
PAR: LI R1, #F0H		8																
SWI R1, 0AH		9																
B CICLO		10																

Tabla 3: Formatos de instrucción del programa del ejemplo 3.

Ejemplo 4. Realizar un programa que compare 3 números colocados en los registros R0, R1 y R2 y guarde el mayor de los 3 en la dirección de memoria 0x20. Considere los 3 números diferentes. Observemos el diagrama de flujo de la figura 4.

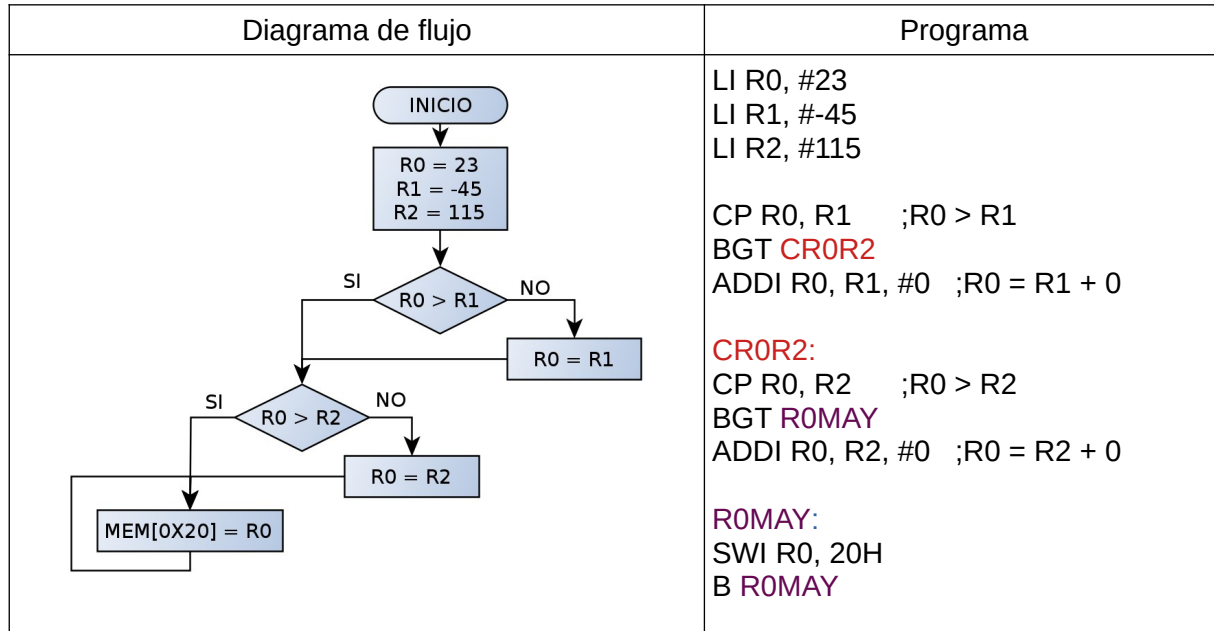


Figura 4: Ejemplo 4 Comparación de números.

El programa que implementa el diagrama de flujo de la figura 4 se muestra en la tabla 4.

Instrucciones	Significado	Dir	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	T
LI R0, #23		0																
LI R1, #-45		1																
LI R2, #165		2																
CP R0, R1		3																
BGT CR0R2		4																
ADDI R0, R1, #0		5																
CR0R2: CP R0, R2		6																
BGT ROMAY		7																
ADDI R0, R2, #0		8																
ROMAY: SWI R0, 20H		9																
B ROMAY		10																

Tabla 4: Formatos de instrucción del programa del ejemplo 4.